**//GOPI K**

**//NISHANTHKUMAR K S**

**//SRIKANTH S**

**/\*WASHING MACHINE SIMULATOR USING   
PICSIMLAB- PIC16F877A\*/**

**//main.c**

#include"main.h"

#pragma config WDTE=OFF

unsigned char reset\_mode,program\_no=0,water\_level\_index=0;

int wash\_time,rinse\_time,spin\_time;

char \*washing\_prog[]={"Daily","Heavy","Delicates","Whites","Stainwash","EcoCottons","Wollens","Bedsheets","Rinse+Dry","Dry only","Wash only","Aqua Store"};

char \*water\_level\_arr[]={"Auto","Low","Medium","High","Max"};

unsigned char min,sec;

unsigned char operation\_mode;

static void init\_config(void)

{

init\_clcd();

init\_digital\_keypad();

init\_timer2();

PEIE=1;

BUZZER\_DDR=0;//buzzer as output

FAN\_DDR=0;//fan as output

GIE=1;//setting Global Interrupt Enable

ADCON1=0x06;//Turn of analog to digital convertors

DC\_MOTOR\_DDR=0;//motor as output

PORTA=OFF;//Initially motor is kept off

}

void main(void)

{

unsigned char key;

operation\_mode = WASHING\_PROGRAM\_DISPLAY;

init\_config();

clcd\_print(" Press Key5 To ",LINE1(0));

clcd\_print(" Power ON ",LINE2(0));

clcd\_print(" Washing Machine ",LINE3(0));

while(read\_digital\_keypad(STATE) != SW5)

{

for(int j=0;j<3000;j++);

}

clear\_screen();// erase all the previous message on screen

power\_on\_screen(); //Calling power on screen

reset\_mode = WASHING\_PROGRAM\_DISPLAY\_RESET;

while(1)

{

key=read\_digital\_keypad(STATE);

for(int j=0;j<3000;j++);

if(key == LSW4 && operation\_mode == WASHING\_PROGRAM\_DISPLAY)

{

operation\_mode=WATER\_LEVEL;

reset\_mode=WATER\_LEVEL\_RESET;

}

else if(key==LSW4 && operation\_mode==WATER\_LEVEL)

{

set\_time\_for\_program();

clear\_screen();// erase all the previous message on screen

clcd\_print(" Press Switch:",LINE1(0));

clcd\_print(" SW5: START ",LINE2(0));

clcd\_print(" SW6: STOP ",LINE3(0));

operation\_mode=START\_STOP\_SCREEN;

}

else if(key == SW5 && operation\_mode ==PAUSE)

{

TMR2ON = ON;//.timer is on

FAN = ON;//turning on fan

DC\_MOTOR=ON;//turning on motor

operation\_mode=START\_PROGRAM;

}

switch(operation\_mode)

{

case WASHING\_PROGRAM\_DISPLAY:

washing\_program\_display(key);

break;

case WATER\_LEVEL:

water\_level\_display(key);

break;

case START\_STOP\_SCREEN:

if(key==SW5)

{

operation\_mode=START\_PROGRAM;

reset\_mode=START\_PROGRAM\_RESET;

continue;

}

else if(key==SW6)

{

clear\_screen();//erases all the previous content on screen

clcd\_print("Returning To ",LINE1(2));

clcd\_print("Main Screen...",LINE2(2));

\_\_delay\_ms(1000);

operation\_mode=WASHING\_PROGRAM\_DISPLAY;

reset\_mode=WASHING\_PROGRAM\_DISPLAY\_RESET;

continue;

}

break;

case START\_PROGRAM:

run\_program(key);

break;

}

reset\_mode=RESET\_NOTHING;

}

}

void power\_on\_screen(void)

{

for(unsigned char i=0;i<16;i++)

{

clcd\_putch(BLOCK,LINE1(i));

}

clcd\_print(" Powering ON ",LINE2(0));

clcd\_print(" Washing Machine ",LINE3(0));

for(unsigned char i=0;i<16;i++)

{

clcd\_putch(BLOCK,LINE4(i));

}

\_\_delay\_ms(1000);

}

void clear\_screen(void)//function that helps to clear the screen

{

clcd\_write(CLEAR\_DISP\_SCREEN,INST\_MODE);

\_\_delay\_us(100);

}

void washing\_program\_display(unsigned char key)

{

if(reset\_mode == WASHING\_PROGRAM\_DISPLAY\_RESET)

{

clear\_screen();// erase all the previous message on screen

program\_no=0;

}

if(key==SW4)

{

program\_no++;

if(program\_no==12)

{

program\_no=0;

}

clear\_screen();// erase all the previous message on screen

}

clcd\_print("Washing Programs",LINE1(0));

clcd\_putch('\*',LINE2(0));

if(program\_no<=9)

{

clcd\_print(washing\_prog[program\_no],LINE2(2));

clcd\_print(washing\_prog[program\_no + 1],LINE3(2));

clcd\_print(washing\_prog[program\_no + 2],LINE4(2));

}

else if(program\_no==10)

{

clcd\_print(washing\_prog[program\_no],LINE2(2));

clcd\_print(washing\_prog[program\_no + 1],LINE3(2));

clcd\_print(washing\_prog[0],LINE4(2));

}

else if(program\_no==11)

{

clcd\_print(washing\_prog[program\_no],LINE2(2));

clcd\_print(washing\_prog[0],LINE3(2));

clcd\_print(washing\_prog[1],LINE4(2));

}

}

void water\_level\_display(unsigned char key)

{

if(reset\_mode==WATER\_LEVEL\_RESET)

{

water\_level\_index=0;

clear\_screen();// erase all the previous message on screen

}

if(key == SW4)

{

water\_level\_index++;

if(water\_level\_index==5)

{

water\_level\_index=0;

}

clear\_screen();// erase all the previous message on screen

}

clcd\_print("Water Level:",LINE1(0));

clcd\_putch('\*',LINE2(0));

if(water\_level\_index <= 2)

{

clcd\_print(water\_level\_arr[water\_level\_index],LINE2(2));

clcd\_print(water\_level\_arr[water\_level\_index + 1],LINE3(2));

clcd\_print(water\_level\_arr[water\_level\_index + 2],LINE4(2));

}

else if(water\_level\_index == 3)

{

clcd\_print(water\_level\_arr[water\_level\_index],LINE2(2));

clcd\_print(water\_level\_arr[water\_level\_index + 1],LINE3(2));

clcd\_print(water\_level\_arr[0],LINE4(2));

}

else if(water\_level\_index == 4)

{

clcd\_print(water\_level\_arr[water\_level\_index],LINE2(2));

clcd\_print(water\_level\_arr[0],LINE3(2));

clcd\_print(water\_level\_arr[1],LINE4(2));

}

}

void set\_time\_for\_program(void)

{

switch(program\_no)

{

case 0: //Daily

switch(water\_level\_index)

{

case 1:

sec=33;

min=0;

break;

case 0:

case 2:

sec=41;

min=0;

break;

case 3:

case 4:

sec=45;

min=0;

}

break;

case 1: //Heavy

switch(water\_level\_index)

{

case 1:

sec=43;

min=0;

case 0:

case 2:

sec=50;

min=0;

break;

case 3:

case 4:

sec=57;

min=0;

}

break;

case 2: //Delicates

switch(water\_level\_index)

{

case 1:

case 0:

case 2:

sec=26;

min=0;

break ;

case 3:

case 4:

sec=31;

min=0;

}

break;

case 3: //Whites

sec=16;

min=1;

break;

case 4: //Stain\_wash

sec=36;

min=1;

case 5: //Eco\_cottons

sec=28;

min=0;

break;

case 6: //Wollens

sec=29;

break;

case 7: //Bed\_sheets

switch(water\_level\_index)

{

case 1:

sec=46;

min=0;

case 0:

case 2:

sec=53;

min=0;

break;

case 3:

case 4:

sec=60;

min=0;

}

break;

case 8:

switch(water\_level\_index)

{

case 1:

sec=18;

min=0;

case 0:

case 2:

case 3:

case 4:

sec=20;

min=0;

}

break;

case 9: //Dry\_Wash

sec=6;

break;

case 10: //Wash\_only

case 11: //Aqua store

switch(water\_level\_index)

{

case 1:

sec=16;

min=0;

case 0:

case 2:

sec=21;

min=0;

break;

case 3:

case 4:

sec=26;

min=0;

}

break;

}

}

void door\_status\_check(void)

{

if(RB0 == 0)//Door open

{

BUZZER=ON;//Turn on th Buzzer

FAN=OFF;//turning off fan

DC\_MOTOR=OFF;//turning off motor

TMR2ON=OFF;//timer is off

clear\_screen();// erase all the previous message on screen

clcd\_print("DOOR : OPEN",LINE1(0));

clcd\_print("Please CLose",LINE2(0));

while(RB0==0)

{

;

}

clear\_screen();// erase all the previous message on screen

clcd\_print("Function -",LINE1(0));

clcd\_print("Time= ",LINE2(0));

clcd\_print("5-START 6-PAUSE",LINE4(0));

FAN=ON;//turning on fan

DC\_MOTOR=ON;

TMR2ON=ON;

BUZZER=OFF;

}

}

void run\_program(unsigned char key)

{

static int total\_time,time;

if(reset\_mode==START\_PROGRAM\_RESET)

{

clear\_screen();// erase all the previous message on screen

clcd\_print("Prog: ",LINE1(0));

clcd\_print(washing\_prog[program\_no],LINE1(5));

clcd\_print("Time: ",LINE2(0));

clcd\_putch((min/10)+'0',LINE2(6));

clcd\_putch((min%10)+'0',LINE2(7));

clcd\_putch(':',LINE2(8));

clcd\_putch((sec/10)+'0',LINE2(9));

clcd\_putch((sec%10)+'0',LINE2(10));

clcd\_print("(MM:SS)",LINE3(5));

\_\_delay\_ms(3000);

clear\_screen();// erase all the previous message on screen

clcd\_print("Function -",LINE1(0));

clcd\_print("Time = ",LINE2(0));

clcd\_print("5-START 6-PAUSE",LINE4(0));

time=total\_time=(min\*60+sec);

wash\_time= (int)(total\_time \* 0.46);

rinse\_time=(int)(total\_time \* 0.12);

spin\_time=(total\_time - wash\_time - rinse\_time);

TMR2ON = 1;//Starting the timer

FAN=ON;//Starting Fan

DC\_MOTOR=ON;//turning on motor

}

door\_status\_check();//checking the status of the door(SW1)

if(key==SW6)//pause the process

{

TMR2ON=OFF;//stops the timer

FAN=OFF;//turning off fan

operation\_mode=PAUSE;

DC\_MOTOR=OFF;//turning off motor

}

total\_time=(min \* 60 + sec);

if(program\_no<7)

{

if(total\_time >=( time - wash\_time))

{

clcd\_print("Wash",LINE1(11));//wash mode

}

else if(total\_time >= (time - wash\_time - rinse\_time))

{

clcd\_print("Rinse",LINE1(11));//rinse mode

}

else

clcd\_print(" Spin",LINE1(11));//spin mode

}

else if(program\_no==8)

{

if(total\_time>=(time - (int)(0.40\*time)))

{

clcd\_print("Rinse",LINE1(11));//rinse mode

}

else

clcd\_print(" Spin",LINE1(11));//spin mode

}

else if(program\_no==9)

{

clcd\_print("Rinse",LINE1(11));

}

else

{

clcd\_print("Wash",LINE1(11));

}

clcd\_putch((min/10)+'0',LINE2(6));

clcd\_putch((min%10)+'0',LINE2(7));

clcd\_putch(':',LINE2(8));

clcd\_putch((sec/10)+'0',LINE2(9));

clcd\_putch((sec%10)+'0',LINE2(10));

if(sec==0 && min==0) //when time up

{

TMR2ON = OFF;//stop the timer

FAN = OFF;//fan is off

DC\_MOTOR=OFF;//turning off the motor

BUZZER = ON;//alert to user using buzzer

clear\_screen();// erase all the previous message on screen

clcd\_print("Prog Completed",LINE1(0));

clcd\_print("Remove Clothes",LINE2(0));

\_\_delay\_ms(1000);

BUZZER=OFF;//turning off the buzzer

operation\_mode=WASHING\_PROGRAM\_DISPLAY;

reset\_mode=WASHING\_PROGRAM\_DISPLAY\_RESET;

clear\_screen();// erase all the previous message on screen

}

}

**//main.h**

#ifndef MAIN\_H

#define MAIN\_H

#include<xc.h>

#include"digitalkeypad.h"

#include"clcd.h"

#include"timer.h"

#define BUZZER RC1

#define BUZZER\_DDR TRISC1

#define FAN RC2

#define FAN\_DDR TRISC2

#define ON 1

#define OFF 0

#define DC\_MOTOR RA0

#define DC\_MOTOR\_DDR TRISA

#define PAUSE 0x05

#define WASHING\_PROGRAM\_DISPLAY 0x01

#define WASHING\_PROGRAM\_DISPLAY\_RESET 0x10

#define RESET\_NOTHING 0x00

#define WATER\_LEVEL 0x02

#define WATER\_LEVEL\_RESET 0x11

#define START\_STOP\_SCREEN 0x03

#define START\_PROGRAM 0x04

#define START\_PROGRAM\_RESET 0x12

void washing\_program\_display(unsigned char key);

void power\_on\_screen(void);

void clear\_screen(void);

void water\_level\_display(unsigned char key);

void set\_time\_for\_program(void);

void run\_program(unsigned char key);

#endif

**//clcd.c**

#include<xc.h>

#include"main.h"

static void init\_display\_controller(void)

{

\_\_delay\_ms(30);

clcd\_write(EIGHT\_BIT\_MODE,INST\_MODE);

\_\_delay\_us(4100);

clcd\_write(EIGHT\_BIT\_MODE,INST\_MODE);

\_\_delay\_us(100);

clcd\_write(EIGHT\_BIT\_MODE,INST\_MODE);

\_\_delay\_us(1);

clcd\_write(CLEAR\_DISP\_SCREEN,INST\_MODE);

\_\_delay\_us(100);

clcd\_write(DISP\_ON\_AND\_CURSOR\_OFF,INST\_MODE);

\_\_delay\_us(100);

}

void init\_clcd(void)

{

CLCD\_DATA\_PORT\_DRR=0x00;

CLCD\_RS\_DDR=0;

CLCD\_EN\_DDR=0;

init\_display\_controller();

}

void clcd\_write(unsigned char byte,unsigned char mode)

{

CLCD\_RS = mode;

CLCD\_DATA\_PORT = byte;

CLCD\_EN=HIGH;

\_\_delay\_us(100);

CLCD\_EN=LOW;

\_\_delay\_us(4100);

}

void clcd\_putch(const char data,unsigned char addr)

{

clcd\_write(addr,INST\_MODE);

clcd\_write(data,DATA\_MODE);

}

void clcd\_print(const char\*str,unsigned char addr)

{

clcd\_write(addr,INST\_MODE);

while(\*str!='\0')

{

clcd\_write(\*str,DATA\_MODE);

str++;

}

}

**//clcd.h**

#ifndef CLCD\_H

#define CLCD\_H

#define CLCD\_DATA\_PORT PORTD

#define CLCD\_DATA\_PORT\_DRR TRISD

#define CLCD\_RS RE2

void clcd\_write(unsigned char byte,unsigned char mode);

void clcd\_putch(const char data,unsigned char addr);

void clcd\_print(const char \*str,unsigned char addr);

void init\_clcd(void);

#define CLCD\_EN RE1

#define CLCD\_RS\_DDR TRISE2

#define CLCD\_EN\_DDR TRISE1

#define EIGHT\_BIT\_MODE 0x33

#define \_XTAL\_FREQ 20000000

#define INST\_MODE 0

#define DATA\_MODE 1

#define HIGH 1

#define LOW 0

#define CLEAR\_DISP\_SCREEN 0x01

#define DISP\_ON\_AND\_CURSOR\_OFF 0x0C

#define LINE1(x) (0x80+x)

#define LINE2(x) (0xC0+x)

#define LINE3(x) (0x90+x)

#define LINE4(x) (0xD0+x)

#define BLOCK 0x2A

#endif

**//timer.c**

#include <xc.h>

void init\_timer2(void)

{

T2CKPS0=1;

T2CKPS1=1;

PR2=250;

TMR2IE=1;

TMR2ON=0;

}

**//timer.h**

#ifndef TIMER\_H

#define TIMER\_H

void init\_timer2(void);

#endif /\* TIMER\_H \*/

**//isr.c**

#include <xc.h>

#include"main.h"

void \_\_interrupt() isr(void)

{

extern unsigned char min,sec;

static unsigned int count=0;

if(TMR2IF==1)

{

if(++count==1250)

{

count=0;

if(sec>0)

sec--;

else if(sec==0 &&min>0)

{

min--;

sec=59;

}

}

TMR2IF=0;

}

}

**//digital.c**

#include<xc.h>

#include"digitalkeypad.h"

void init\_digital\_keypad()

{

KEYPAD\_PORT\_DRR=KEYPAD\_PORT\_DRR|ALL\_RELEASED;

}

unsigned char read\_digital\_keypad(unsigned char mode)

{

static unsigned int once=1;

static unsigned char pre\_key;

static unsigned char longpress;

if(mode==LEVEL\_DETECTION)

return (KEYPAD\_PORT & INPUT\_LINES);

else

{

if(((KEYPAD\_PORT & INPUT\_LINES)!=ALL\_RELEASED)&& once)

{

once=0;

longpress=0;

pre\_key=KEYPAD\_PORT & INPUT\_LINES;

}

else if(!once && (pre\_key==(KEYPAD\_PORT & INPUT\_LINES))&& longpress < 20)

{

longpress++;

}

else if(longpress == 20)

{

longpress++;

return 0x80|pre\_key;

}

else if((KEYPAD\_PORT & INPUT\_LINES)==ALL\_RELEASED && !once)

{

once=1;

if(longpress < 20)

{

return pre\_key;

}

}

}

return ALL\_RELEASED;

}

**//digitalkepad.h**

#ifndef DIGITALKEYPAD\_H

#define DIGITALKEYPAD\_H

#define ALL\_RELEASED 0x3F

#define INPUT\_LINES 0x3F

#define LEVEL 0

#define STATE 1

#define KEYPAD\_PORT PORTB

#define LEVEL\_DETECTION 0

#define KEYPAD\_PORT\_DRR TRISB

#define SW1 0x3E

#define SW2 0x3D

#define SW3 0x3B

#define SW4 0x37

#define SW5 0x2F

#define SW6 0x1F

#define LED1 RD0

#define LSW4 0xB7

#define LSW5 0xAF

unsigned char read\_digital\_keypad(unsigned char mode);

void init\_digital\_keypad(void);

#endif /\* DIGITALKEYPAD\_H \*/

THANK YOU

\*\*\*\*\*